

CMPT 280

Topic 7: Cloning

Mark G. Eramian

University of Saskatchewan

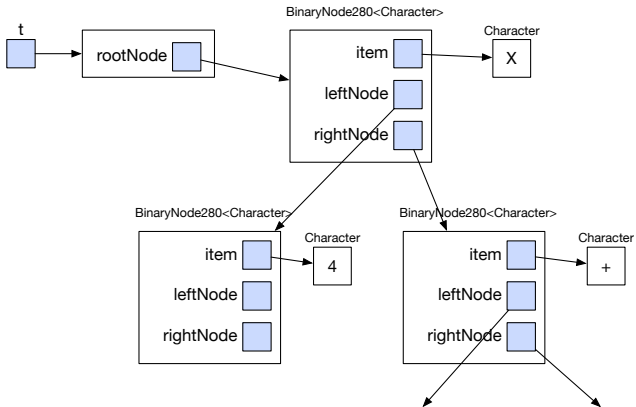
References

- Textbook, Chapter 7

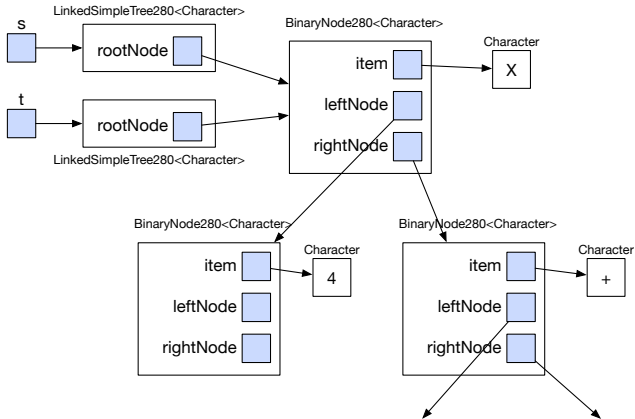
Shallow vs Deep Clone

- Shallow clone copies only the object's data and references.
- Deep clone copies all of the other objects reference directly or indirectly by the object being cloned.
- Suppose `t` is a reference to a `LinkedSimpleTree<Character>` object and we write:
`LinkedSimpleTree<Character> s = t.clone()`

Shallow Clone (before)

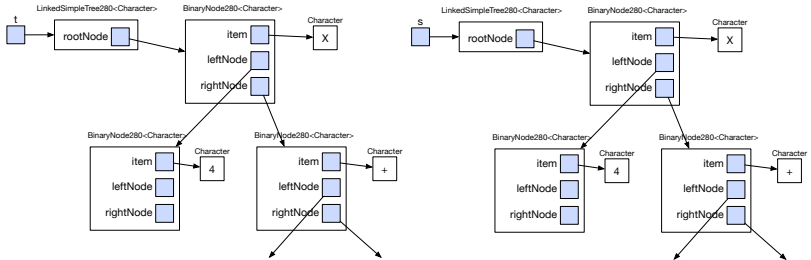


Shallow Clone (after)



Only the object that was cloned gets duplicated.

Deep Clone (after)



The object cloned gets duplicated, as well as all objects it directly or indirectly refers to.

Cloning in Java

- In Java, the `Object` class has a method called `clone()` which can make a shallow clone of the object.
- It would seem, then, that every Java class supports shallow cloning automatically.
- But let's take a closer look at the **documentation for `Object`**...

Cloning in Java

- Whoops – clone() is a protected method in Object! That means another object can't call clone(). So what good is it?
- It is protected because Java insists that you grant explicit permission that it is safe to shallow clone your object by just copying its fields.
- This permission is granted by having your objects implement the Cloneable interface. Let's take a look...

Cloning in Java

- Classes that implement `Cloneable` must implement a public method called `clone`.
- This can either override the `clone` method in `Object`, or call it explicitly (which is allowed, because protected methods can be called by members of the same class or its descendants!).

Exercise 1

- Make `BinaryNode280<I>` cloneable (shallow clone).
- Make `SimpleTree280<I>` cloneable (shallow clone).

Deep Clones in Java

- Deep clones are enabled in Java in much the same way as shallow clones.
- Instead of calling the protected `clone` method in `Object` from our public `clone` method, we write our own custom code, appropriate to the data structure, to construct a deep clone.
- But before we can demonstrate how to deep-clone our binary tree, we need to know about tree traversals.

Next Class

- Next class reading: Chapter 8: Tree Traversals.